

Mohne – Mobile Drohne

Abschlussbericht

Entwicklung eines autonomen Fahrzeuges für extraterrestrische Erkundungsmissionen

Eingereicht bei:

**Fachhochschule Kufstein Tirol Bildungs GmbH**

**Studiengang Smart Products & Solutions**

Verfasser:

**Markus Wiesmüller**

**1710809534**

Abgabedatum:

**11.03.2018**

Table of Contents

[List of Figures IV](#_Toc508553330)

[List of Tables V](#_Toc508553331)

[List of Abbreviation VI](#_Toc508553332)

[Kurzfassung VII](#_Toc508553333)

[Abstract VIII](#_Toc508553334)

[1. System Beschreibung 1](#_Toc508553335)

[1.1 Mechanisches Konzept 1](#_Toc508553336)

[1.1.1 Flugmodus 1](#_Toc508553337)

[1.1.2 Fahrmodus 3](#_Toc508553338)

[1.1.3 Funktionsprinzip der Propeller 3](#_Toc508553339)

[1.2 Sensing Concept 4](#_Toc508553340)

[1.2.1 Picamera 5](#_Toc508553341)

[1.2.2 Ultraschallabstandssensor 5](#_Toc508553342)

[1.2.3 Gesamtsystem 6](#_Toc508553343)

[2. Materialien und Methoden 7](#_Toc508553344)

[2.1 Materialien 8](#_Toc508553345)

[2.2 Manufacturing Technologies 8](#_Toc508553346)

[2.3 Software Tools 10](#_Toc508553347)

[2.3.1 Autodesk Inventor Professional 11](#_Toc508553348)

[2.3.2 Autodesk EAGLE 12](#_Toc508553349)

[2.3.3 Arduino IDE 13](#_Toc508553350)

[2.3.4 Raspbian 13](#_Toc508553351)

[2.3.5 Python 14](#_Toc508553352)

[2.3.6 Samba 15](#_Toc508553353)

[3. System Life Cycle 16](#_Toc508553354)

[3.1 System Requirements 16](#_Toc508553355)

[3.1.1 Mandatory Requirements 16](#_Toc508553356)

[3.1.2 Preference Requirements 16](#_Toc508553357)

[3.1.3 Limitations 17](#_Toc508553358)

[3.2 Design Alternatives 17](#_Toc508553359)

[3.3 Model the System 18](#_Toc508553360)

[3.3.1 System and Subsystem 18](#_Toc508553361)

[3.3.2 Make or Bye 18](#_Toc508553362)

[3.3.3 System Architecture 19](#_Toc508553363)

[3.3.4 Failure Mode and Effects Analysis (FMEA) 20](#_Toc508553364)

[3.4 Integrate System Components 21](#_Toc508553365)

[3.5 Launch the System 22](#_Toc508553366)

[3.5.1 Timeline 22](#_Toc508553367)

[3.5.2 Budget 23](#_Toc508553368)

[3.6 Assess Performance 24](#_Toc508553369)

[3.7 Results, Conclusion and Recommendations 25](#_Toc508553370)

[3.7.1 Results 25](#_Toc508553371)

[3.7.2 Conclusion 26](#_Toc508553372)

[3.7.3 Recommendations 26](#_Toc508553373)

[4. Diagrams 27](#_Toc508553374)

[4.1 Use Case Diagram 27](#_Toc508553375)

[4.2 Structural Model 28](#_Toc508553376)

[4.3 Interaction Model 29](#_Toc508553377)

[4.4 Behavior Model 29](#_Toc508553378)

[5. Version Control 29](#_Toc508553379)

[6. Design of PCB 30](#_Toc508553380)

[6.1 Breadboard 30](#_Toc508553381)

[6.2 Schematic 30](#_Toc508553382)

[6.3 Board 31](#_Toc508553383)

[7. Lessons Learned 32](#_Toc508553384)

# ****List of Figures****

[Abbildung 1: Quadrocopter: Motordrehrichtung 2](#_Toc508553385)

[Abbildung 2: Quadrocopter: Vorwärtsbewegung 2](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553386)

[Abbildung 3: Quadrocopter: Rückwärtsbewegung 2](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553387)

[Abbildung 4: Tragfläche mit Propeller: Verstellbar um +90° und -90° 3](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553388)

[Abbildung 5: Quadrocopter: Mechanisches Konzept 4](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553389)

[Abbildung 6: Angelernte Formen und Farben 5](#_Toc508553390)

[Abbildung 7: Ultraschallabstandssensor 6](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553391)

[Abbildung 8: Funktionsprinzip Ultraschallabstandssensor 6](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553392)

[Abbildung 9: Quadrocopter: Sensorkonzept 7](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553393)

[Abbildung 10: CAD-Modell des Hybrid-Quadrocopters 11](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553394)

[Abbildung 11: System and Subsystems 18](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553395)

[Abbildung 12: System architecture Mohne 19](#_Toc508553396)

[Abbildung 13: Integration of subsuystems into the overall system 21](#_Toc508553397)

[Abbildung 14:Project Timeline 22](#_Toc508553398)

[Abbildung 15: Use Case Diagram 27](#_Toc508553399)

[Abbildung 16: Structural Model 28](#_Toc508553400)

[Abbildung 17: Interaktion Model 29](file:////Users/Markus/Google%20Drive/20180212_CaseStudyReport_v1-3.docx#_Toc508553401)

# ****List of Tables****

[Tabelle 1: Zur Verfügung gestelltes Material 8](#_Toc508553402)

[Tabelle 2: Werkzeug und Maschinen 8](#_Toc508553403)

[Tabelle 3: Komponentenübersicht 12](#_Toc508553404)

[Tabelle 4: Make or Bye Decision 19](#_Toc508553405)

[Tabelle 5: FMEA 20](#_Toc508553406)

[Tabelle 6: Project Tasks and Responsibility 22](#_Toc508553407)

[Tabelle 7: Budget Calculation 23](#_Toc508553408)

[Tabelle 8: Test Tracks 24](#_Toc508553409)

# ****List of Abbreviation****

|  |  |
| --- | --- |
| **IoT** | **Internet of Things** |
| **GM** | **Geschäftsmodell** |
| **GMI** | **Geschäftsmodell-Innovation** |
| **DBT** | **Digital Business Transformation** |
| **IT** | **Informationstechnik** |
| **EY** | **Ernst & Young** |
| **ROI** | **Return on Investment** |
| **KPI** | **Key Performance Indicator** |
| **MIT** | **Massachusetts Institute of Technologie** |
| **IKT** | **Informations- und Kommunikationstechnik** |
| **HR** | **Human Resources** |
| **BDI** | **Bundesverband der Deutschen Industrie e.V.** |
| **CEO** | **Chief Executive Officer** |
| **F&E** | **Forschung und Entwicklung** |

# Kurzfassung

In diesem Jahrhundert werden Menschen zum ersten Mal ihre Füße auf den Planeten Mars setzen – zumindest sehen dies durchaus realistische Pläne diverser Raumfahrtorganisationen. Bis es jedoch soweit ist, müssen weitere Projekte wie Phoenix oder InSight durchgeführt werden, um weitere Informationen über die Entstehung des Mars zu sammeln. Die Aufgabe, welche daher für das Projekt gestellt wurde, war die Entwicklung eines autonom agierenden Fahrzeugs, welches den widrigen Bedingungen auf dem Mars standhalten würde. Nach reichlicher Überlegung kamen wir zu dem Entschluss, dass eine Hybriddrohne, welche sowohl fahren als auch fliegen kann am besten für die Erkundungsmission geeignet sei, da dadurch die Möglichkeit gegeben ist, große Höhenunterschiede wie Berge oder Schluchten mühelos zu bewältigen und einen erheblichen Vorteil gegenüber konventionellen Mars-Rovern darstellt. Um diese Annahme zu bestätigen, hat das Team unterschiedliche Designvariationen in Betracht gezogen, bevor wir uns entschlossen haben einen Hybrid-Quadrocopter zu bauen. Die Nutzlast unserer Drohne umfasst eine Kamera und einen Ultraschallsensor, wodurch der Drohne ermöglicht wird sich autonom fortzubewegen. Hinsichtlich Projektmanagementmethodik entschieden wir uns für einen traditionellen Ansatz für die Gesamtplanung und für eine agile Methodik für die Einzelaufgaben.

Um die benötigten Komponenten für die Drohne zu produzieren wurden vom Fablab in Wattens die entsprechenden Maschinen und Geräte sowie hilfreiche Unterstützung zur Verfügung gestellt.

Um festzustellen, ob der Mars-Rover für einen Einsatz auf dem Mars geeignet ist, mussten fünf verschiedene Tracks absolviert werden. Es stellte sich jedoch schnell heraus, dass ein stabiler Flug und somit ein überwinden von Hindernissen aufgrund der fehlenden PID-Regelung nicht möglich war.

Letztendlich war dieses Konzept in der vorgegebenen Zeit für uns nicht realisierbar, weshalb man sich besser für ein anderes Design entschieden hätte. Man sollte sich daher nicht für ein Konzept entscheiden, von welchem man überzeugt ist, welches mit den gegebenen Ressourcen jedoch nicht umsetzbar ist. Dennoch zeigte unsere Drohne, dass sie bei weiterer Entwicklung das Potential für einen Einsatz auf dem Mars hat.

# Abstract

In this century, humans will set their feet on the planet Mars for the first time - at least this is what realistic plans of various space organisations see. Until then, however, further projects such as Phoenix or In-Sight will have to be carried out in order to gather further information on the origin of Mars. The task, therefore, was to develop an autonomous vehicle that would withstand the adverse conditions on Mars. After a lot of discussions, we came to the conclusion that a hybrid drone, which can drive as well as fly, is best suited for exploration missions, as it offers the possibility to effortlessly cope with big differences in altitude like mountains or canyons and represents a considerable advantage compared to conventional Mars rovers. To confirm this assumption, the team considered different design variations before we decided to build a hybrid quadrocopter. The payload of our drone includes a camera and an ultrasonic sensor, which enables the drone to move autonomously. With regard to project management methodology, we opted for a traditional approach to overall planning and an agile methodology for the individual tasks.

In order to produce the required components for the drone, the Fablab in Wattens provided the appropriate machines and equipment as well as helpful support.

In order to determine whether the Mars rover is suitable for a mission on Mars, five different tracks had to be completed. However, it soon became clear that a stable flight and thus the overcoming of obstacles was not possible due to the lack of PID regulation.

Ultimately, this concept was not feasible for us in the given time, which is why it would have been better to opt for a different design. Therefore, you should not decide on a concept that you are convinced of, but which cannot be implemented with the given resources. Nevertheless, our drone showed that it has the potential to be used on Mars during further development.

# System Beschreibung

## Mechanisches Konzept

Die Mohne ist ein sogenannter „Hybrid-Quadrocopter“, der sowohl zum Fahren als auch zum Fliegen konzipiert ist. Durch diese Hybrid-Nutzung ergeben sich große Vorteile. Auf dem Boden spart der Roboter Energie (im Verhältnis zur Fortbewegung in der Luft), während er in der Luft auch meterhohe Hindernisse überwinden kann. Nachfolgend wird das mechanische Konzept beider Varianten näher erläutert sowie in Abbildung 5 veranschaulicht.

### Flugmodus

Die Mohne gehört zur Gruppe der Quadrocopter, bestehend aus vier Motoren die in H-Konfiguration angeordnet sind. Bei der H-Konfiguration sind die vier Motoren in 45° zur Flugrichtung versetzt, wobei der Abstand vom Kreuzmittelpunkt zu den einzelnen Motoren immer gleich groß ist.

Vier Propeller sind fest an den Motoren montiert und erzeugen den Auftrieb der Drohne, welcher sich über die Drehzahl der einzelnen Motoren verändern lässt. Übersteigt die durch die sich drehenden Propeller erzeugte Auftriebskraft die Anziehungskraft der Erde bzw. des Mars, so beginnt die Drohne zu schweben. Eine Erhöhung bzw. Verringerung der Drehzahl hat ein Steigen bzw. ein Sinken der Drohne zufolge.

Dadurch ergibt sich ein relativ einfacher mechanischer Aufbau, in dem es außer den Rotoren der Motoren keine weiteren beweglichen Teile für den Flugmodus gibt.

Überkreuz liegende Rotoren drehen dabei in die gleiche Richtung (Abbildung 1) und heben somit das um die Mitte der Drohne übertragene Drehmomente auf.

Bei exakt gleicher Drehgeschwindigkeit aller Propeller und keinen äußeren Störeinflüssen, steht die Drohne also exakt lotrecht über dem Boden. Dies eliminiert die Notwendigkeit einer Gierstabilisierung.

Um ein nach vorne fliegen zu ermöglichen, drehen sich die vorderen Rotoren langsamer als die hinteren, was ein nach vorne abkippen der Drohne zur Folge hat (Abbildung 2). Entsprechend wird ein nach hinten fliegen ermöglicht, indem die hinteren Rotoren langsamer als die vorderen drehen (Abbildung 3).

Abbildung 1: Quadrocopter: Motordrehrichtung

A

B

C

D

y

x

A

B

C

D

z

Abbildung 2: Quadrocopter: Vorwärtsbewegung

A

B

C

D

z

Abbildung 3: Quadrocopter: Rückwärtsbewegung

### Fahrmodus

Um mit der Mohne auch fahren zu können, wurde eine Radaufhängung mit Achsen an die Drohne montiert. Allerdings werden dabei die Räder nicht von einem Motor betrieben, stattdessen wird die Schubleistung von den Propellern der Mohne erzeugt. Die hinteren Tragflächen des Quadrocopters lassen sich im Fahrmodus über Zahnräder um nahezu 90° nach vorne und hinten verstellen (Abbildung 4). In diesem Modus drehen sich nur die hinteren Propeller, wodurch ein Vorschub ähnlich wie bei einem Flugzeug bzw. Schiff erzeugt wird. Ein Abbremsen erfolgt indem die Propeller auf Umkehrschub gestellt werden. Sie erzeugen so einen nach vorne gerichteten Schub. Die Verstellung erfolgt dabei elektrisch.

+90°

-90°

Abbildung 4: Tragfläche mit Propeller: Verstellbar um +90° und -90°

### Funktionsprinzip der Propeller

Wichtig für das Verständnis des Funktionsprinzips der Drohne ist, dass man sich grob phänomenologisch die Wirkungsweise von Propellern veranschaulicht, weil sich schon aus recht simplen Überlegungen Hinweise darauf ergeben, wie die Propeller sinnvollerweise ausgelegt werden müssen. Die Wirkungsweise eines Propellers, dessen Aufgabe sowohl im Flug- als auch im Fahrmodus darin besteht eine Drehleistung von optimal in eine Schubleistung umzusetzen, kann aus zwei Anschauungen gewonnen werden, die dann in jeweils unterschiedlichen Berechnungsmodellen münden:

* **Impulsbetrachtung:** Um einen Schub T nach vorne bzw. nach oben zu erzeugen, muss der Propeller die Luft nach hinten bzw. nach unten beschleunigen. Die Änderung des Impulses entspricht dabei dem Propellerschub. Die Theorie, die auf diesem Modell aufbaut, ist die Strahltheorie.
* **Tragflügelbetrachtung:** Die Wirkungsweise eines Propellers entspricht der eines Tragflügels. Der Auftrieb entspricht dabei dem Schub, der Widerstand dem Moment. Das Tragflügelmodell erklärt die Propellerwirkung an den einzelnen Flügelschnitten und ist als Traglinentheorie (nur radiale Diskretisierung) oder Tragflächentheorie bekannt.

Aus diesen beiden Wirkungsweisen lassen sich schon einige grundlegende Dinge qualitativ ableiten, ohne dass genaue mathematische Begründungen nötig sind.

Abbildung 5: Quadrocopter: Mechanisches Konzept

**Motor**

**+**

**Propeller**

**Motor**

**+**

**Propeller**

**Motor**

**+**

**Propeller**

**Motor**

**+**

**Propeller**

**Rad**

**Rad**

**Rad**

**Rad**

**Servo**

**Servo**

## Sensing Concept

Um die Mohne autonom fliegen bzw. fahren zu lassen und Hindernisse und Ziele zu erkennen, wurde eine Kamera und ein Ultraschallsensor verbaut. Nachfolgend werden beide Sensoren näher beschrieben und das System im Gesamtkontext betrachtet.

### Picamera

Mit der Kamera sollen vorher definierte Formen und Farben erkannt werden. Die Rückgabewerte an den Raspberry Pi sollen in der Logik die vorgesehene Funktion auslösen. Beispiele hierfür sind markierte Hindernisse und das Ziel. Anbei ein Auszug an angelernten Formen und Farben (Abbildung 6), hier ist auch das vorgegebene rote Quadrat enthalten. Die exakte Größe spielte keine Rolle, da wir in der Logik Ecken und Kanten zählen haben lassen und Farben erkennen.



Abbildung 6: Angelernte Formen und Farben

### Ultraschallabstandssensor

Mit dem Ultraschallabstandssensor (Abbildung 7) kann man mittels eines Ultraschalllautsprechers und eines Mikrofons den Abstand berührungslos zu einem Objekt messen. Das Prinzip (Abbildung 8) basiert darauf, dass die Schallgeschwindigkeit in der Luft bei gleichbleibender Temperatur nahezu konstant bleibt - bei 20°C beträgt sie 343,2m/s. Aus diesem Fakt kann man die Abstandsmessung in eine Zeitmessung überführen, welche dann von Mikro-Controllern einfach übernommen werden kann.

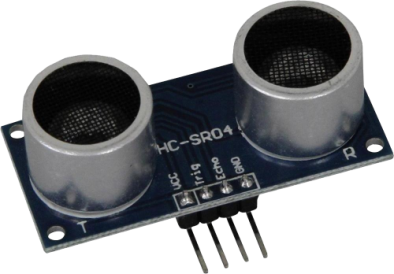


Abbildung 7: Ultraschallabstandssensor

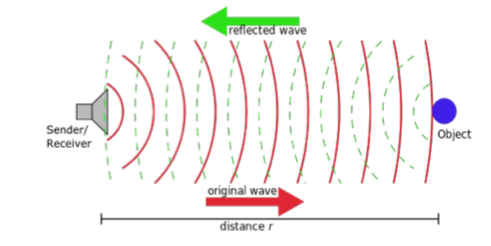


Abbildung 8: Funktionsprinzip Ultraschallabstandssensor

### Gesamtsystem

Die unter Punkt 1.2.1 und 1.2.2 beschriebenen Sensoren unterstützen den Hybrid-Quadrocopter sich in seiner Umgebung zurechtzufinden und sich autonom fortzubewegen, Ziele zu erkennen und Hindernisse gekonnt zu überfliegen. Die Picamera dient dazu ein Ziel zu erkennen und dieses anzusteuern, mit Hilfe des Ultraschallsensors kann ein Hindernis erkannt werden, woraufhin in den Flugmodus gewechselt wird, um das Hindernis zu überfliegen und um nach einer bestimmten Zeit x zu landen und wieder in den Fahrmodus zu wechseln. Wie in Abbildung 9 veranschaulicht sind dabei beide Sensoren mit einem Micro-Controller verbunden.

Abbildung 9: Quadrocopter: Sensorkonzept

**Controller**

**Kamera**

**Ultraschallsensor**

# Materialien und Methoden

Um das Projekt zu realisieren und die Mohne zu entwickeln, wurde Seitens des Auftraggebers verschiedene Materialien und Werkzeugmaschinen sowie ein Budget von 40€ zur Verfügung gestellt.

Es gab viele Faktoren, die dazu beitrugen, wie dieser Hybrid-Quadrocopter konstruiert werden sollte. Das Team musste zuerst berücksichtigen, dass die Drohne Testflüge absolvieren musste, von denen sie gegebenenfalls nicht in einem Stück zurückkehrt. Hinzu kam, dass das Gewicht der für die Herstellung der Drohne verwendeten Materialien sorgfältig geprüft werden musste, damit das Gewichtslimit und auch die Größe für den Wettbewerb nicht überschritten wurde. Schließlich musste das Design auf eine Art und Weise erstellt werden, welches leicht mit den zur Verfügung stehenden Materialien und Werkzeugmaschinen umgesetzt werden konnte. Zudem musste das Design die Elektronik vor Wasser schützen. Aufgrund des entscheidensten Faktors, die für die Produktion zur Verfügung stehenden Zeit, wurde entschieden, Lasercutten, 3D-Druck, Schneiden, Schleifen, Fräsen und verbinden mit Schrauben zu verwenden. Dieser Prozess wurde als am schnellsten und am leichtesten reproduzierbar bestimmt, falls erforderlich. Der erste Schritt in diesem Prozess war das Design der Drohne zu entwerfen. Dafür wurde die in Abschnitt 2.3.1 beschriebene CAD-Software Autodesk Inventor verwendet. Im nächsten Schritt wurde bestimmt, wie die einzelnen Komponenten hergestellt und welche Materialen dafür verwendet werden sollten. Komponenten wir Arduino, Raspberry Pi und Sensoren wurden zur Verfügung gestellt. Da jedoch kein Breadboard verwendet werden durfte, wurde mit Hilfe der in 2.3.2 beschriebenen Software ein Leiterplatten-Layout erstellt, welches mit Hilfe einer Fräsmaschine gefertigt wurde.

In diesem Kapitel werden die bereitgestellten Materialien aufgelistet, die verfügbaren Maschinen vorgestellt sowie eine Vorstellung der verwendeten Software.

## Materialien

Aus der Tabelle 1 sind die Materialien und Maße der vom Kunden zur Verfügung gestellten Werkstoffe ersichtlich.

Tabelle 1: Zur Verfügung gestelltes Material

|  |  |
| --- | --- |
| **Material** | **Maße in mm (l x b x h)** |
| Holz | 800 x 600 x 3,8 |
| Plexiglas | 600 x 400 x 3 |

## Manufacturing Technologies

Um das Projekt zu realisieren kamen unterschiedliche Werkzeugmaschinen zum einsatz, welche aus Tabelle 2 ersichtlich sind. Auf eine nähere Beschreibung der Handhabung der Maschinen wird in diesem Bericht bewusst verzichtet.

Tabelle 2: Werkzeug und Maschinen

|  |  |
| --- | --- |
| **Werkzeug / Maschine** |  |
| Bohr-Fräsmaschine Optimum BF 16 Vario |  |
| Schraubstock |  |
| Schleifmaschine Holzstar BTS 150 |  |
| Bandsäge Metabo BAS 317 Precision |  |
| Digitaler Messschieber Scala 150mm |  |
| Ultimaker 2 3D-Drucker |  |
| Ultimaker 3 Dual Extruder 3D-Drucker |
| Fräsmaschine Roland SRM-20 |  |
| Lötstation Weller WS 81 |  |
| Lasergravierer Trotec Speedy 360 |  |
| Metallsäge |  |
| Schraubenzieher |  |
| Schraubenschlüssel |  |

## Software Tools

Um das Projekt zu realisieren, wurden die nachfolgenden Softwareprogramme verwendet. Zum einen wurde Autodesk Inventor Professional verwendet, um ein 3D CAD-Modell der Mohne zu erstellen (Abbildung 10). Die einzelnen Komponenten sind in Tabelle 3 aufgelistet. Zum anderen wurde für die Konstruktion der Leiterplatte das Programm EAGLE der Firma Autodesk verwendet. Weitere Software Tools wurden für die Programmierung benötigt.

### Autodesk Inventor Professional

Autodesk Inventor ist eine von Autodesk entwickelte 3D-CAD-Software für die Volumenmodellierung, mit der digitale 3D-Prototypen erstellt werden können. Es wird für 3D-Konstruktion, Design-Kommunikation, Werkzeugherstellung und Produktsimulation verwendet. Diese Software ermöglicht es Benutzern, genaue 3D-Modelle zu erstellen, um Produkte zu entwerfen, zu visualisieren und zu simulieren, bevor sie gebaut werden.

Die Software ermöglicht zudem 2D- und 3D-Daten in eine einzige Designumgebung zu integrieren und so eine virtuelle Darstellung des Endprodukts zu erstellen, mit der die Form, Passform und Funktion des Produkts vor dem Bau validieren werden kann. Autodesk Inventor enthält leistungsstarke parametrische, direkte Bearbeitungs- und Freiform-Modellierungswerkzeuge sowie Multi-CAD-Übersetzungsfunktionen und DWG Zeichnungen nach Industriestandard, wodurch Entwicklungskosten gesenkt, schnellere Time-to-Market realisiert und großartige Produkte entwickelt werden können.

Abbildung 10: CAD-Modell des Hybrid-Quadrocopters

Tabelle 3: Komponentenübersicht

|  |  |
| --- | --- |
| **Name** | **Anzahl** |
| Chassis mit motorlöcher | 1 |
| Platzhalter Zylinder Klein | 4 |
| Metall Stab M5 | 1 |
| Chassis-2 | 1 |
| Kegelrad | 1 |
| EMax Motor | 4 |
| RaspberryPi2 STEP214 | 1 |
| Metall Stab M5\_MIR | 1 |
| StirnradUNDServo\_MIR | 1 |
| Propeller | 4 |
| arduino uno | 1 |
| MPU 6050 IMU | 1 |
| Abstandhalter\_MPU | 1 |
| LiPo Akku | 1 |
| PDB | 4 |
| Ultrasonic Sensor Halterung | 1 |
| USS | 1 |
| Wheel ALL | 2 |
| Radaufhängung | 4 |
| Wheel ALL1\_MIR | 2 |
| Achse | 2 |
| Kamerahalterung | 1 |
| ESC | 4 |
| M3 Schrauben | 27 |
| M3 Beilagscheiben | 27 |
| M3 Muttern | 27 |
| Raspberry Pi Camera | 1 |
| Stirnzahnrad1 | 2 |
| Stirnzahnrad2 | 2 |
| servo\_SM-S2309S | 2 |

### Autodesk EAGLE

EAGLE ist eine skriptfähige EDA-Anwendung (Electronic Design Automation) mit Schaltplan-Erfassung, Leiterplatten-Layout, Auto-Router und CAM-Funktionen (Computer Aided Manufacturing). EAGLE steht dabei für Easy Applicable Grafischer Layout-Editor.

Das Programm bietet eine grafische Benutzeroberfläche und ein Menüsystem mit mehreren Fenstern für die Bearbeitung, das Projektmanagement und die Anpassung der Schnittstellen- und Designparameter. Das System kann über Maus, Tastaturkürzel oder durch Eingabe bestimmter Befehle in einer eingebetteten Befehlszeile gesteuert werden. Mehrere Wiederholungsbefehle können zu Skriptdateien kombiniert werden. Es ist auch möglich, Entwurfsdateien unter Verwendung einer EAGLE-spezifischen objektorientierten Programmiersprache zu untersuchen.

### Arduino IDE

Arduino bringt eine eigene integrierte Entwicklungsumgebung (IDE) mit, die auf Wiring IDE basiert. Dabei handelt es sich um eine Java-Anwendung, die für die gängigen Plattformen Windows, Linux und MacOS kostenlos verfügbar ist. Sie basiert auf der IDE von Processing, einer auf die Einsatzbereiche Grafik, Simulation und Animation spezialisierten Entwicklungsumgebung. Die Arduino-IDE bringt einen Code-Editor mit und bindet gcc als Compiler ein. Zusätzlich werden die avr-gcc-Library und weitere Arduino-Librarys eingebunden, die die Programmierung in C und C++ stark vereinfachen. Arduino kann verwendet werden, um eigenständige interaktive Objekte zu steuern oder um Softwareanwendungen auf Computern zu interagieren (z.B. Adobe Flash, Processing, Max/MSP, Pure Data, diverse Skriptsprachen, etc.)

Für ein funktionstüchtiges Programm genügt es, zwei Funktionen zu definieren:

* setup() – wird beim Start des Programms (entweder nach dem Übertragen auf das Board oder nach Drücken des Reset-Tasters) einmalig aufgerufen, um z. B. Pins als Eingang oder Ausgang zu definieren.
* loop() – wird durchgehend immer wieder durchlaufen, solange das Arduino-Board eingeschaltet ist.

### Raspbian

Raspbian ist ein kostenloses Betriebssystem, das auf Debian basiert und für die Raspberry Pi-Hardware optimiert ist. Ein Betriebssystem ist die Menge der grundlegenden Programme und Dienstprogramme, die den Raspberry Pi laufen lassen. Raspbian bietet jedoch mehr als ein reines Betriebssystem: Es enthält mehr als 35.000 Pakete, vorkompilierte Software in einem schönen Format für die einfache Installation auf dem Raspberry Pi. Dieses Betriebssystem basiert auf einem Debian-9-System(Debian Stretch) der ARM-hard-float-Architektur (armhf) mit Anpassungen an den Befehlssatz für den ARMv6-Prozessor. Als grafische Oberfläche wird LXDE vorkonfiguriert. Das etwa 3 GB große Image kann auf SD-Karten mit 4 GB oder mehr übertragen werden. Nach dem Bootvorgang kann die Größe der Raspbian-Partition auf die gesamte SD-Karte erweitert werden. Die Raspberry Pi Foundation erstellt auf Basis der Raspbian-Distribution ein eigenes Raspbian-Image mit passender Firmware für die Raspberry-Pi-Modelle, es wird daher empfohlen, die Distribution immer von der Raspberry Pi Foundation zu beziehen.

### Python

Das Pi im Namen des Raspberry Pi leitet sich von der Programmiersprache Python ab: Die Schöpfer des Minicomputers wollten damit zum Ausdruck bringen, dass diese moderne Skriptsprache besonders zu empfehlen ist.

Python ist eine universelle, üblicherweise interpretierte höhere Programmiersprache. Sie hat den Anspruch, einen gut lesbaren, knappen Programmierstil zu fördern. So werden beispielsweise Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert.

Python unterstützt mehrere Programmierparadigmen, z. B. die objektorientierte, die aspektorientierte und die funktionale Programmierung. Ferner bietet es eine dynamische Typisierung. Wie viele dynamische Sprachen wird Python oft als Skriptsprache genutzt.

Die Sprache hat ein offenes, gemeinschaftsbasiertes Entwicklungsmodell, das durch die gemeinnützige Python Software Foundation, die de facto die Definition der Sprache in der Referenzumsetzung CPython pflegt, gestützt wird.

Wegen ihrer klaren und übersichtlichen Syntax gilt Python als einfach zu erlernen. Sie besitzt eine umfangreiche Standardbibliothek und zahlreiche Pakete im Python Package Index.

Zur Standardinstallation von Python gehört eine integrierte Entwicklungsumge- bung namens IDLE.

Raspbian bringt die Python-Entwicklungsumgebung IDLE bereits vorinstalliert auf dem Desktop mit, sodass bei Bedarf direkt losgelegt werden kann.

IDLE besteht im Wesentlichen aus drei Komponenten:

* **Die Python-Shell:** Beim Starten von IDEL öffnet sich zuerst das Python-Shell- Fenster. Die Shell ist eine Anwendung, mit der Sie direkt mit dem Python-In- terpreter kommunizieren können: Sie können auf der Kommandozeile einzel- ne Python-Anweisungen eingeben und ausführen lassen. Ein Python-Pro- gramm, das eine Bildschirmausgabe liefert, gibt diese in einem Shell-Fenster aus.
* **Der Programmeditor:** Das ist eine Art Textverarbeitungsprogramm zum Schreiben von Programmen. Sie starten den Programmeditor vom Shell-Fens- ter aus (FILE|NEW WINDOW).
* **Der Debugger:** Er dient dazu, den Lauf eines Programms zu kontrollieren, um logische Fehler zu finden.

### Samba

Sabba ist ein freies Programmpaket, das es ermöglicht Windows-Funktionen wie die Datei- und Druckdienste unter anderen Betriebssystemen zu nutzen und die Rolle eines Domain Controllers anzunehmen. Es implementiert hierfür unter anderem das Server Message Block/Common Internet File System-Protokoll (SMB/CIFS-Protokoll).

Da die Software unter der General Public Licence (GPL) kostenfrei verfügbar ist, wird sie als Alternative zu Microsoft-Windows-Server-Betriebssystem eingesetzt.

# System Life Cycle

Within the scope of the Mars Explorer project, the knowledge gained from the courses Systems Engineering, Embedded Systems and Project Management will be applied. In addition, a Fablab environment, mechanical and electronic component manufacturing and PCB production will be explored.

## System Requirements

### Mandatory Requirements

The aim of the project is to develop an autonomous vehicle suitable for extraterrestrial exploration missions with the provided components, materials, machinery and budget. The suitability is tested on five test tracks. The requirements for the vehicle are as following:

* Undamaged fall from a maximum drop height of 0.5 m
* Protection against dust and water
* Dimensions for rocket transport:
  + Lenght < 65 cm
  + Width < 48 cm
  + Height < 30 cm
* Weight oft he vehicle:
  + mRover < 10 kg (Earth)
  + mRover < 4,5 kg (Mars)
* Type of movement freely selectable, but autonomous
* Budget: 40 €

### Preference Requirements

The project should lead to an autonomous hybrid quadrocopter, which can overcome the tracks described under 3.6 by changing between driving and flight mode and recognizing its environment by means of an ultrasonic sensor and camera and responding accordingly to different situations. The heart of the vehicle is a Raspberry Pi and an Arduino which receive, process and transmit data.

Sub-goals for the project are:

* Design and manufacture the chassis
* Design and manufacture the rotary mechanism to switch between ride and flight modes
* Design and manufacture printed circuit boards (PCB)
* Programming of the engine control incl. ESC calibration to create the conditions for flying
* Autonomous mode using sensors (programming for ultrasonic sensor and camera)
* Control code and PIC control to keep the Quadrocopter stable

### Limitations

Switching on and off takes place in manual mode, while the Mohne is operated autonomous during flying and driving. The distance to the ground is not measured. Overcoming obstacles is achieved by changing from driving mode to flight mode and returning to driving mode after a pre-set time.

The system has to work only under good weather conditions and on given courses, therefore only a straight flight is realized.

## Design Alternatives

In order to decide upon a design for this project, certain limiting factors had to be taken into account, namely the maximum size and maximum permissible weight that the vehicle may have in order to be still be able to be transported by the launch vehicle to Mars. Even under the condition that the electronic components, which would be incased and secured somewhere inside the fuselage, must be protected from dust and water, the possibilities were further restricted. The given budget and time limit, as well as the materials and machines available, limited further possibilities. After some thought and research was carried out to find the easiest way to overcome the individual hurdles of the test tracks, it was decided that a hybrid quadrocopter would be developed. Due to the fact that drones are becoming increasingly popular and are being used in an increasing number of applications, we wanted to meet this challenge and abandon the construction of a conventional Mars rover.

## Model the System

Normally, models are developed for almost all alternative designs. The preferred alternative model is then expanded and used to manage the system throughout its entire system lifecycle. Due to the short development time, no models were developed for alternative designs. First and foremost, the following models are also intended to describe the system.

### System and Subsystem

The overall system was partitioned into four subsystems (Abbildung 11) which made it easier to design, understand and test the system and which made it easier to work in teams.

Abbildung 11: System and Subsystems

### Make or Bye

In a make-or-buy decision, the most important factors to consider are part of quantitative analysis, such as the associated costs of production and whether the business has the capacity to produce at required levels. As a result, it was relatively easy to decide which parts to buy and which to produce. The production of the electronic components could not be realized by us, which is why they were purchased, but the mechanical components were designed and manufactured by us (Tabelle 4).

Tabelle 4: Make or Bye Decision

|  |  |
| --- | --- |
| Motor | Commercially available subsystem |
| Propeller | Commercially available subsystem |
| ESC | Commercially available subsystem |
| Servo | Commercially available subsystem |
| Gyro and USS | Commercially available subsystem |
| Arduino | Commercially available subsystem |
| Raspberry Pi | Commercially available subsystem |
| Carrier | In-house-design |
| Chassis | In-house-design |
| Rotary mechanism | In-house-design |
| Battery | Commercially available subsystem |
| Wires | Commercially available subsystem |
| Power Distribution Board | First Make, after “burn out” Buy |
| Wiring | Commercially available subsystem |
| Sensors | Commercially available subsystem |

### System Architecture

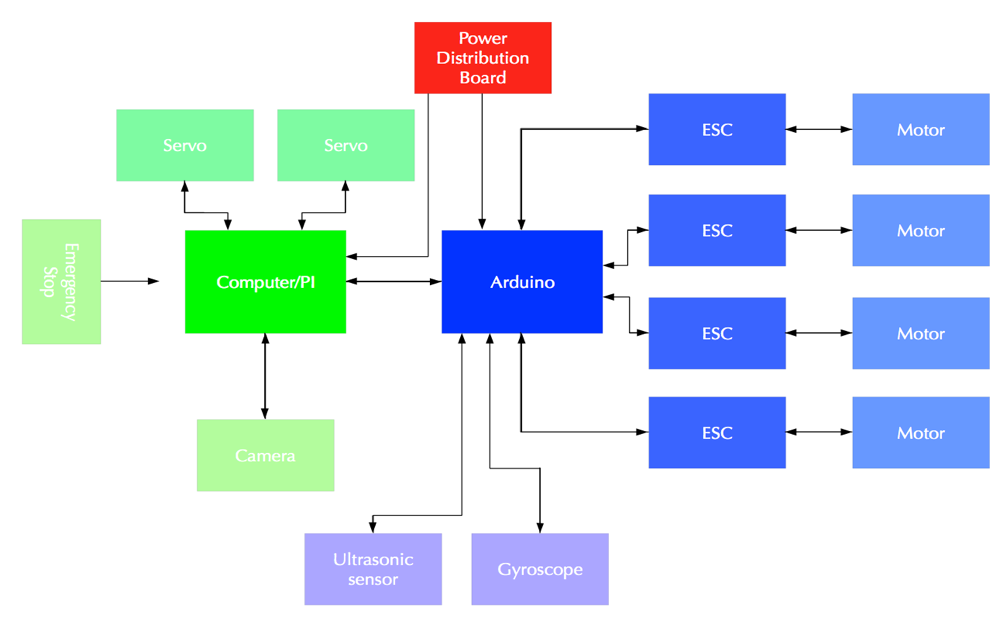


Abbildung 12: System architecture Mohne

### Failure Mode and Effects Analysis (FMEA)

The procedure used to evaluate the various modes of failure in our drone design takes into consideration the severity, probability of occurrence and detectability for each failure mode as well as the root causes. Based on the values assigned to each category, the risk priority and criticality of each cause are calculated. Our failure analysis focused on four failure modes; mobility, navigation, structural and data acquisition (Tabelle 5). We determined the key failures modes to be mobility and navigation. In each of these two cases, control system and wiring failure are the root causes with the highest risk priorities and criticalities.

Tabelle 5: FMEA

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Possible Effect** | **Root Cause** | **Potential Indicators** | **Potential Cause(s) of Failure** | **Severity** | **Probability of Occurence** | **Detectability** | **Risk Prirority** | **Criticality** |
| **Mobility** | dead battery | will not power motor | Battery not charged | 5 | 3 | 1 | 15 | 15 |
|  | controls failure | Motors, servos and PID controller failure | Programming error | 5 | 9 | 9 | 405 | 45 |
|  | wiring harness connections | will not start or run correctly | Grounding corrosion | 8 | 8 | 9 | 576 | 64 |
| **Navigation** | Wiring failure | Complete navigation failure | Wiring error | 8 | 7 | 2 | 112 | 56 |
|  | Controls Failure | Complete navigation failure | Programming error | 5 | 9 | 9 | 405 | 45 |
|  | Object sensing hardware failure | Part damage | Wiring control failure | 5 | 3 | 3 | 45 | 15 |
|  | Rotary mechanism failure | Steering failure | Impact on gear | 5 | 5 | 1 | 25 | 25 |
| **Structural** | Frame cracking | No component support | Impact on frame | 10 | 3 | 4 | 120 | 30 |
|  | Engin mount | Engin falls out | Engin mount wear | 2 | 1 | 1 | 2 | 2 |
| **Data Acquisition** | Camera not working | No video recording | Weather proofing controls/wire | 1 | 5 | 2 | 10 | 5 |
|  | Ultrasonic sensor not working | No distance measurement | Weather proofing controls/wire | 1 | 5 | 3 | 15 | 5 |
|  | Incorrect wiring | Will not move | Wiring error | 2 | 2 | 4 | 16 | 4 |
|  | Controls failure | No/incorrect data | Programming error | 1 | 2 | 5 | 10 | 2 |

## Integrate System Components

For system integration, an important point is that each part is well developed for itself and somehow works alone. This makes it easier to repair things because it is easier to repair a small part than the whole system. The second important point of system integration is the combination of all subsystems into a fully functional system. Picture 13 fits perfectly to show the integration of subsystems into the overall system "Mohne". The parts of the puzzle that were not part of our project were crossed out and yet there are many parts that we had to combine. Two parts of the puzzle which caused difficulties due to time constraints. This will be explained later, but clearly shows the importance of well integrated system components.

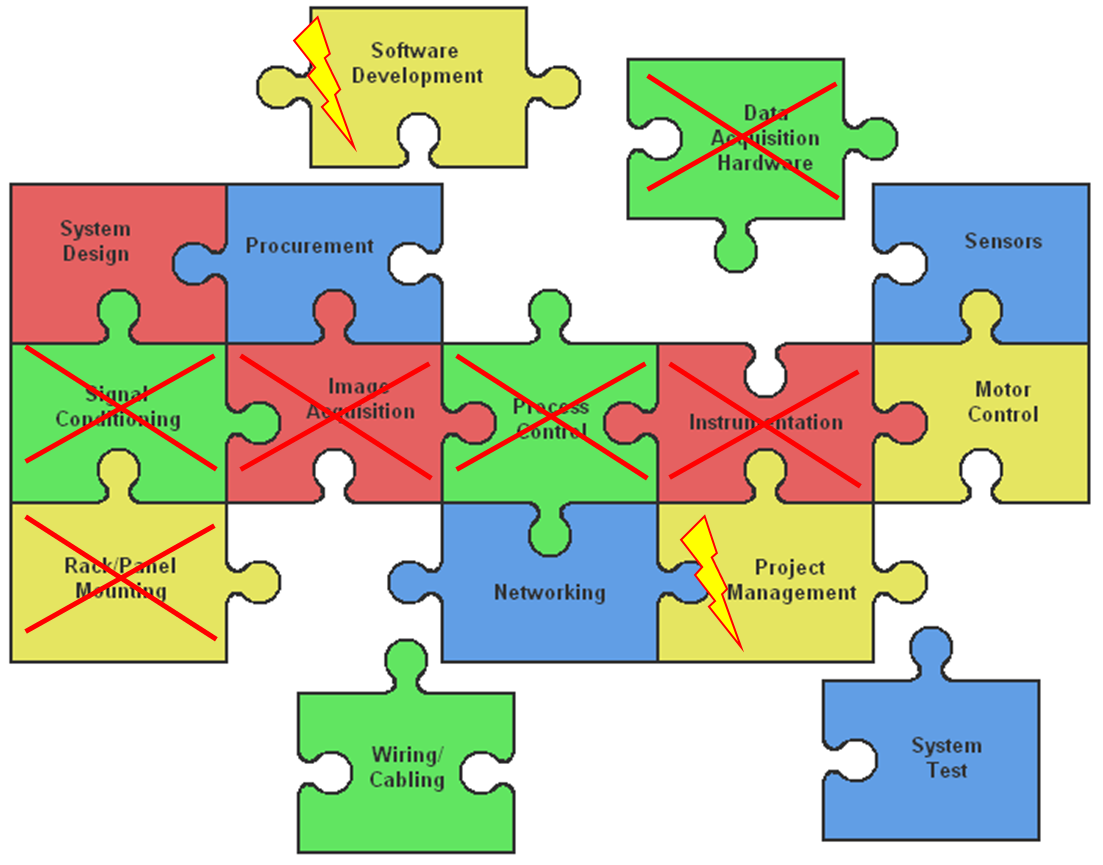


Abbildung 13: Integration of subsuystems into the overall system

## Launch the System

### Timeline

For the overall timeline (Abbildung 14) it was decided to use a traditional planning Method “waterfall”. Project phases were structured, discussed and described and milestones with goals to be reach after each phase were set up.

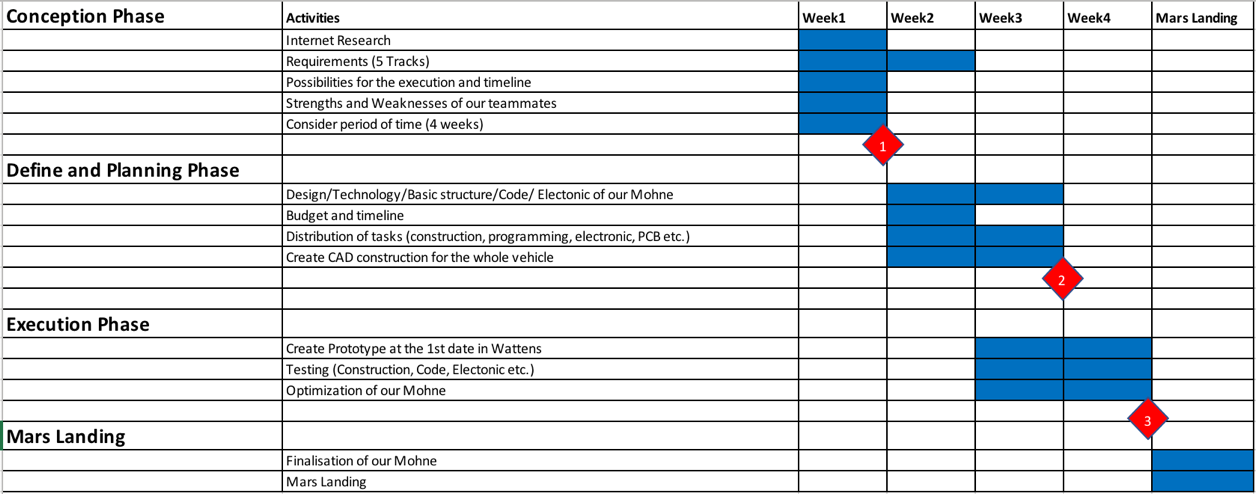


Abbildung 14:Project Timeline

Therefore, tasks (Tabelle 6) were set up in the agile planning tool on github. Traditional planning methods (overall planning) were combined with agile project methods (scrum mode doing the tasks).

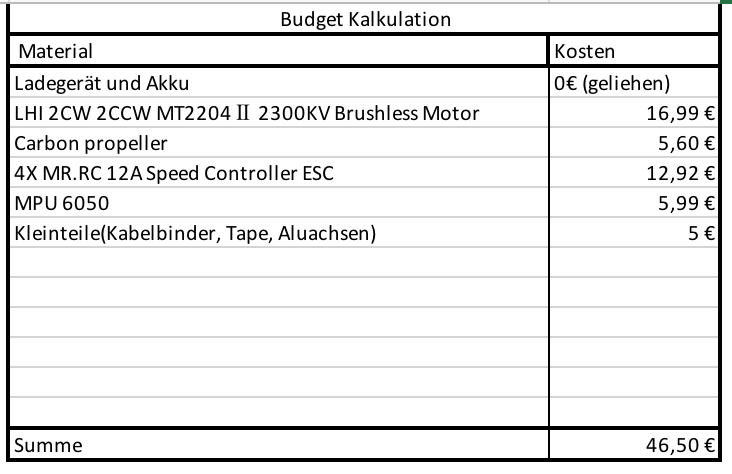
Tabelle 6: Project Tasks and Responsibility

|  |  |
| --- | --- |
| **Task** | **Team Member(s)** |
| Design decision | all |
| Design and manufacture chassis | Duregger, Jungwirth, Gruber |
| Image recognition | Paula, Gruber |
| Camera mounting | Duregger, Jungwirth |
| Suspension, axles and shaft | Duregger, Jungwirth, Gruber |
| PCB layout | Möbes |
| Electronics | Möbes, Gruber |
| Servo mount + rotary mechanism | Duregger, Jungwirth, Gruber |
| Engine mount | Duregger, Jungwirth, Gruber |
| Spacer between boards | Duregger, Jungwirth, Gruber |
| ESC calibration and motor control | Möbes, Gruber |
| Plug connection HV battery and power board | Möbes |
| Control code | Möbes, Paula, Gruber |
| Project management and documentation | Wiesmüller |
| Assembly | Wiesmüller, Gruber |
| Cost analysis | Gruber |
| Component selection and ordering | all |
| Flight and system test | all |

### Budget

Because of the very small budget of 40€ a little budget planning had to be done. Unfortunately, the project could not be realized with the available budget, which is why the budget was a bit overdrawn. For this small budget the project was much too ambitious.

Tabelle 7: Budget Calculation



## Assess Performance

During the entire development phase, the performance of the system was measured and tested again and again. These measurements were necessary to verify that the system met its requirements. First of all, it was tested at the beginning whether the rotary mechanism works as expected, which was the basis for the change between driving and flying mode.

Tests were also carried out again and again in order to ensure the image recognition and distance measurement to ensure autonomous navigation. Another important point that has been tested continuously is the engine control to ensure that the quadcopter can move.

In order to determine whether the project is suitable for a mission on Mars, whether it can move autonomously and fulfills the expectations and wishes of the client, 5 test tracks (Tabelle 8) had to be completed, which are described below. In addition, the clients inspected whether the required minimum requirements in terms of size, weight and stability were met.

Tabelle 8: Test Tracks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Track** | **Dimensions (l x w)** | **Surface** | **Environment condition** | **Goal** |
| 1 | 3m x 1m | Concrete | Indoor area | Drive straight on |
| 2 | 3m x 1m | Concrete | Dust simulation by means of spray bottles | passing through without defects |
| 3 | 3m x 1m | Wooden steps with different heights (10cm, 20cm, 30cm, 40cm, 50cm, 60cm) | Indoor area | Overcoming obstacles |
| 4 | 3m x 1m | Rocks with different heights (10-60 cm) | Indoor area, rocks colored and marked with symbol | Cross the track, avoid obstacles |
| 5 | unbekannt | Unknown (snow, ice, sand, etc.), rocks with different heights (10-60cm) | Outdoor area, rocks colored and marked with symbol | Survive |

Image 15 shows the symbol the rocks are marked with

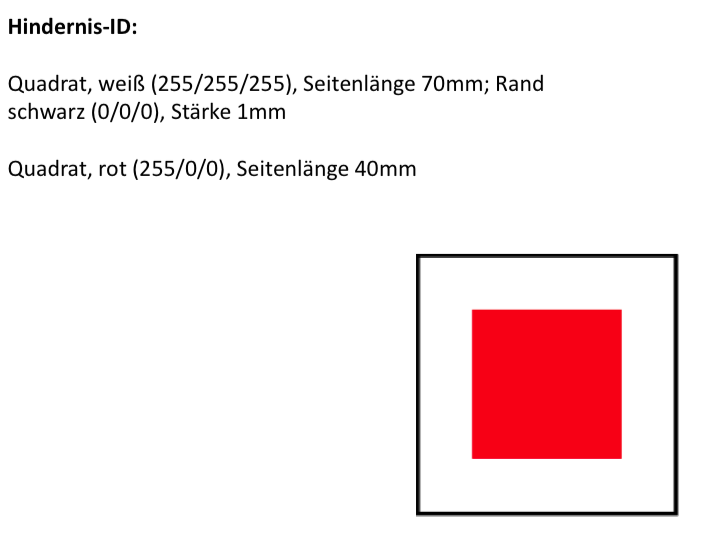


Abbildung 15: Obsticle-ID

## Results, Conclusion and Recommendations

### Results

This project lasted 4 weeks and involved a lot of research, testing and analysis. At the beginning of the project, the team created initial project specifications as detailed in section 3.1 which served as guidelines for the Mohne. A review was done to see which specifications the project fulfilled and which ones it did not.

* The drone met the maximum size and weight requirements
* The drop test from a height of 0.5m was passed unscathed
* The drone met the requirements for dust and water resistance
* The drone failed to meet the requirement to not exceed 40 € in cost. The total cost of the drone was 46,50 €
* The drone met the emergency stop requirement
* The drone met the requirement to detect its environment
* The change between flight and driving mode could be realized
* The drone was able to drive straight on
* The requirement to fly was only partially met. Although several flight tests were carried out, it could not be realized that all four propellers were supplied with consistent electricity distribution, consequently the drone flight was not stable enough. The problem was detected, but the PID control required for this could not be implemented due to time constraints.
* Because of the previous problem, the obstacles could not be overcome

### Conclusion

The final results showed that not all specifications and goals set at the beginning of the project were met. Although the fully autonomous flight was not achieved, the team made a significant development towards creating a low-cost, lightweight unmanned hybrid vehicle capable of surveillance with its implementation of a camera system and an ultrasonic sensor. Several design iterations were completed to create a vehicle that would meet the project goals and the final prototype demonstrated potential for success with the project goals. As described earlier in section 3.6, several tests were performed to ensure the designed prototype would fly and ride successfully. If more time was available, the team would have been able to design a more robust control system that would ensure stable flight. Overall, an unmanned aircraft vehicle was fully realized in this project. The hybrid quadcopter demonstrated it was able to drive and fly, but it lacked stability. Further iterations of this project would ensure its future success.

### Recommendations

Several lessons were learned during the course of this project and thus the team has several recommendations. From a controls aspect, more sensors would be helpful for more precision or capabilities. Another ultrasonic sensor could be used to determine the altitude. The implementation of additional camera modules could create a 360-degree perception to avoid obstacles. In the next step it could also be realized to drive and fly curves. The execution of simulations and calculations would lead to a more effective design.

# Diagrams

Nachfolgend werden Use Case Diagram, Structural Model, Interaction Model und Behavior Model für ein besseres Verständnis des Softwaresystems, welche Komponente eine andere beeinflusst und in welcher Form dies geschieht, wie die Kommunikation zwischen Raspberry Pi und Arduino gestaltet ist sowie das Verhalten eines Systems, entwickelt und beschrieben.

## Use Case Diagram

Das gezeigte Use Case Diagram (Abbildung 16) zeigt die Interaktionen zwischen externen Akteuren (User, PiCam, USS, GYRO, Controller) und dem betrachteten System (Autonome Fortbewegung). Während der User nur für das Starten und den Not-Aus der Mohne zuständig ist, sind die anderen Akteure, sowie die Steuerung der Motoren und Servos, für die Steuerung der Drohne zuständig. Darüber hinaus sind PiCam, Ultraschallsensor und Gyro für die Umgebungserkennung zuständig und Leiten die gemessenen Werte an den Controller weiter.

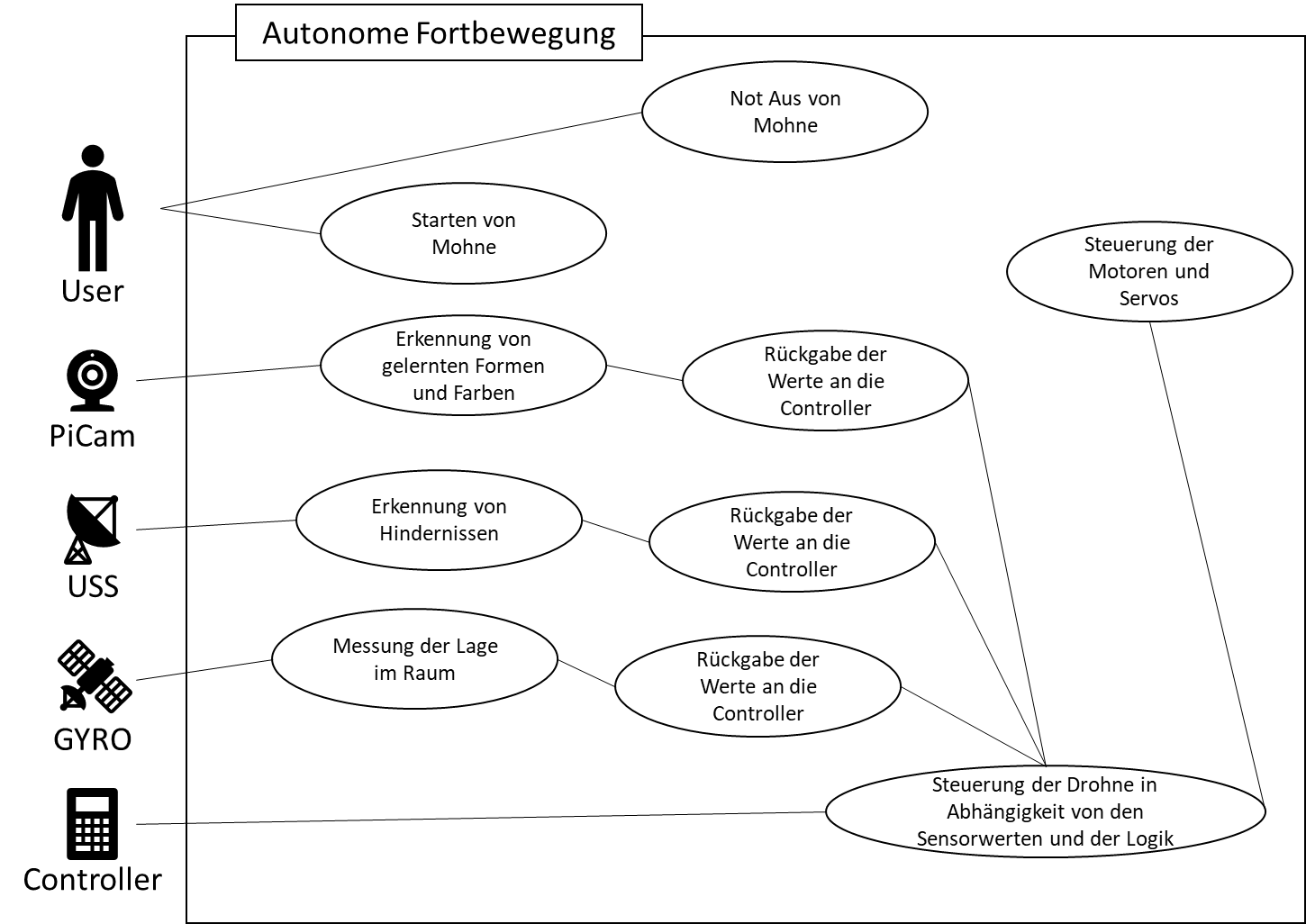


Abbildung 16: Use Case Diagram

## Structural Model

Die Mohne ist ein Fluggerät, welches durch das Zusammenspiel von verschiedenen Faktoren und Systemkomponenten bestimmt wird. Die Abhängigkeit der einzelnen Systemteile wird in der nachstehenden Graphik (Abbildung 17) dargestellt. Durch die dauerhafte Kommunikation untereinander, kann eine Überwachung und Kontrolle der einzelnen Komponenten sichergestellt werden. Dabei stehen die einzelnen Systemteile in einer direkten Abhängigkeit. Im Zentrum der Struktur stehen hier der Arduino Uno bzw. der Raspberry Pi 3. Diese verarbeiten die eingehenden Daten des Gyroskops, des MPU´s, des Ultraschallsensors und der Kamera. Durch die Datenerfassung des der genannten Eingangssysteme werden die Motoren bzw. die Motorleistung entsprechend gesteuert und angepasst. Die erfassten Daten werden vom Gyroskope und MPU an Arduino/Raspberry Pi weitergegeben und diese steuern bzw. überwachend die Motorleistung und passen diese entsprechend an, damit die „Mohne“ kontrolliert fliegen kann. Ebenfalls werden die Servoelemente der „Mohne“ durch den Arduino und Rasberry Pi gesteuert und kontrolliert, welche benötigt werden um sich am Boden vorwärts bewegen zu können.

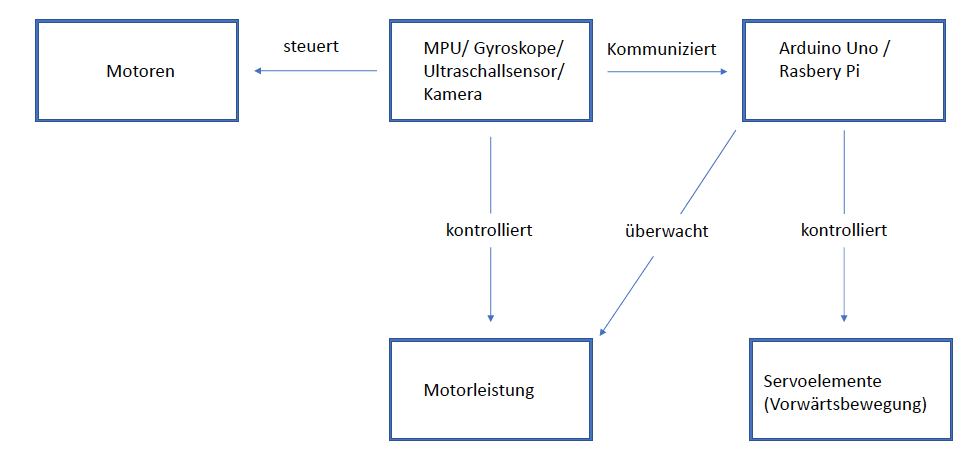


Abbildung 17: Structural Model

## Interaction Model

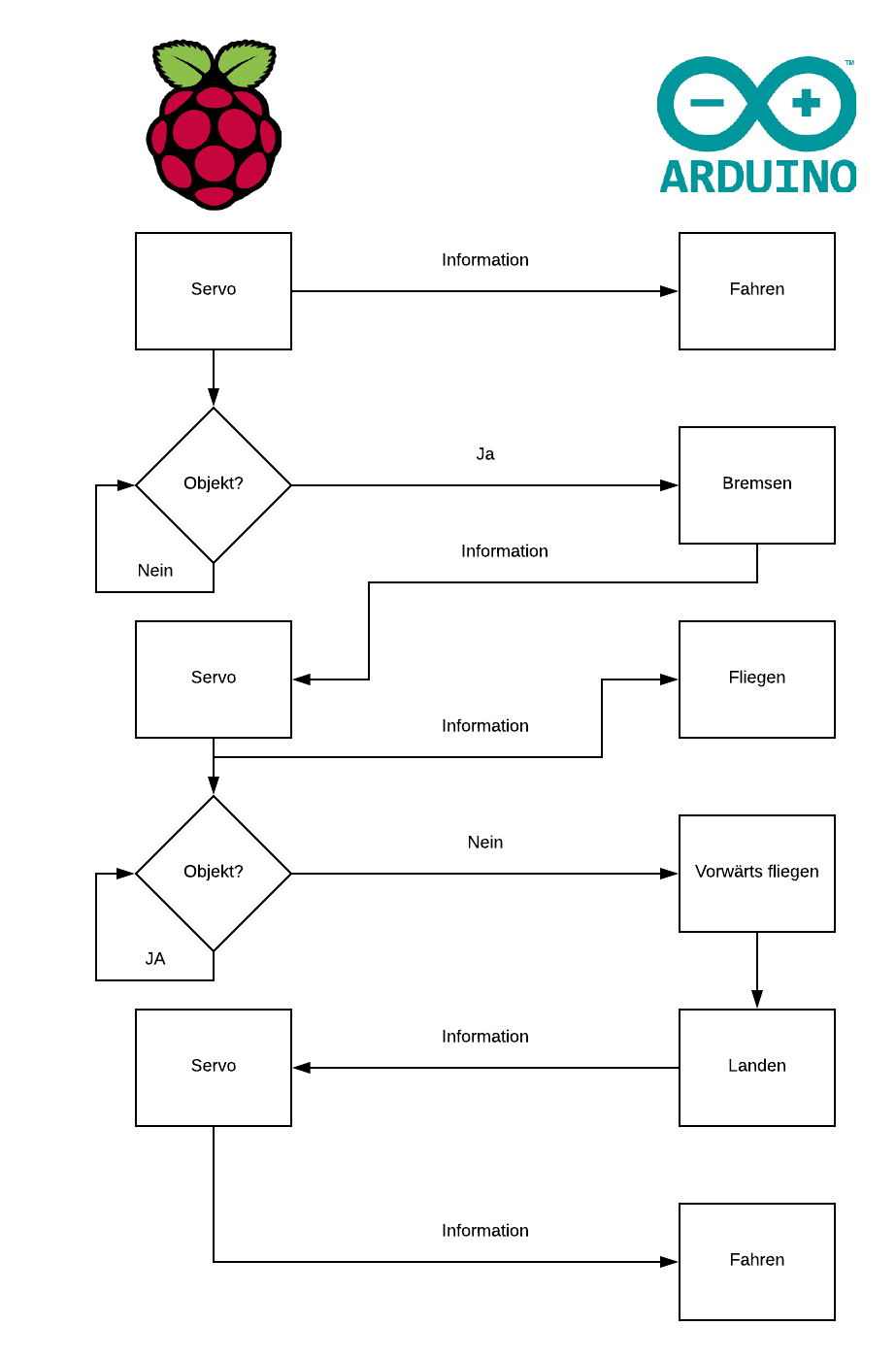


Abbildung 18: Interaction Model

## Behavior Model

# Version Control

Abbildung 19: Behavior Model

<https://github.com/gchrizZz/RTW>

# Design of PCB

## Breadboard

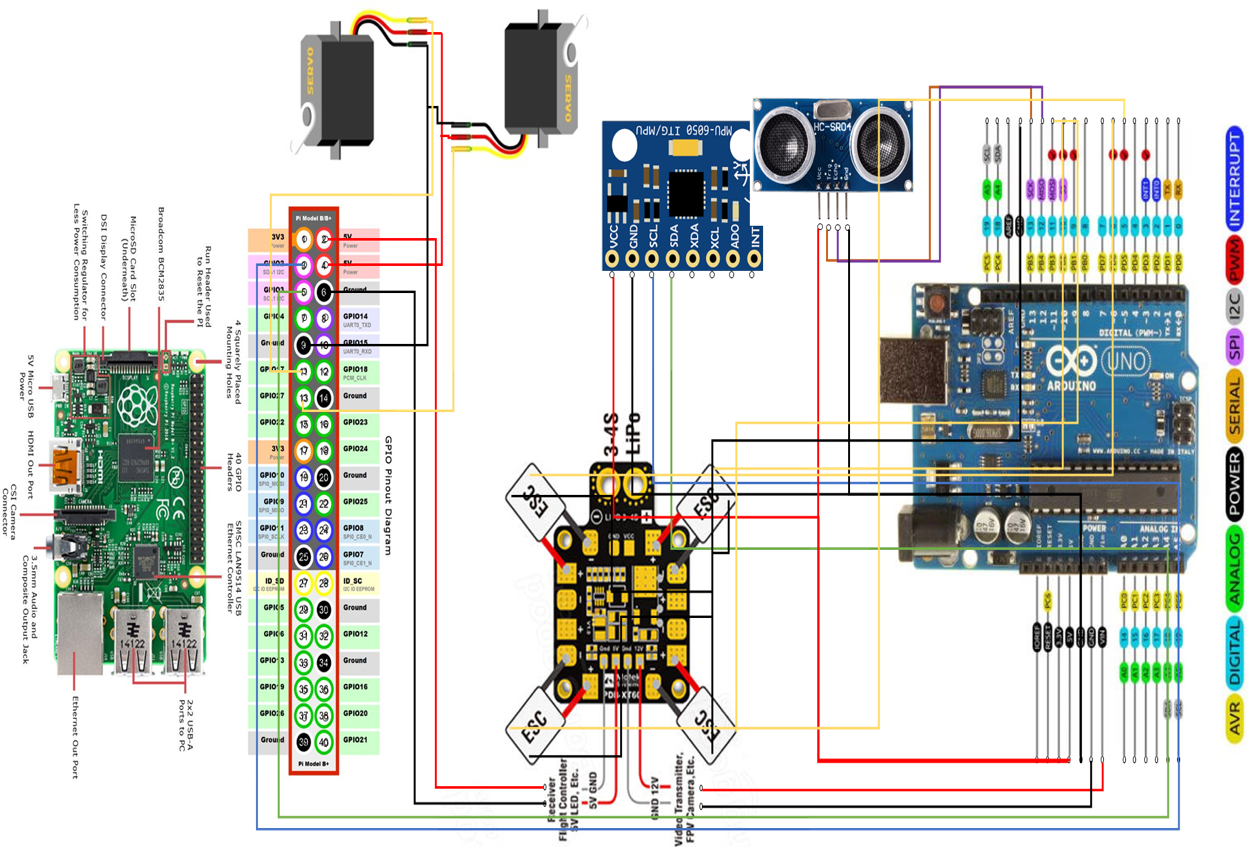


Abbildung 20: Beadboard

## Schematic

Nach einigen erfolgten Diskussionen im Projektteam fiel die Entscheidung auf das Anfertigen einer Platine im Stile eines PCBs. Der ausschlaggebende Faktor für diese Entscheidung war die Prämisse, den Schaltkreis und die letztendliche Verkabelung so einfach wie möglich zu halten, um die Fehleranfälligkeit zu reduzieren.

Die gesamte Konstruktionsarbeit für die Platine erfolgte mit dem Programm Eagle von Autodesk. Der Vorteil bei der Benutzung dieses Programms besteht darin, dass im ersten Schritt die Logik komplett unabhängig vom späteren Layout erstellt werden kann. Die entsprechenden Komponenten (Schraubklemmen) des Herstellers Würth werden aus dessen Onlinebibliothek in Eagle importiert und per Drag & Drop auf das Bearbeitungsfeld gezogen. Im nächsten Schritt werden mithilfe von Labels die einzelnen Leiterbahnen so definiert, dass GND und VCC (Ground und Power) mit der richtigen Logik verbunden sind.

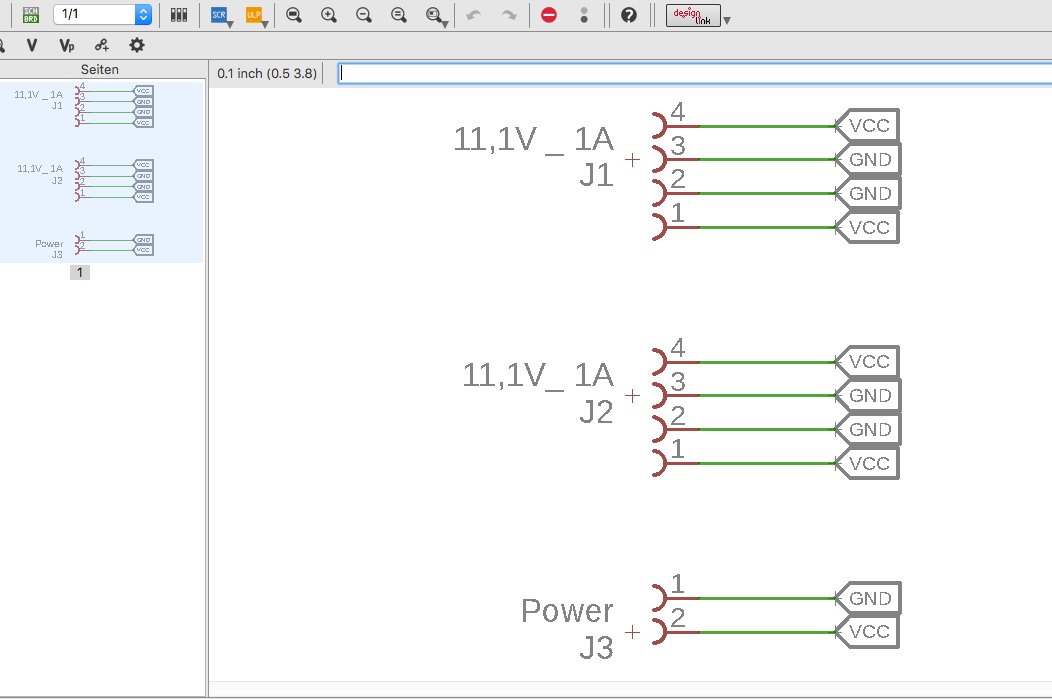


Abbildung 21: PCB-Schematic

## Board

Nach der Erstellung der schematischen Zeichnung gelangt man durch die Aktivierung einer Schaltfläche in die Board-Layout-Ansicht. Hier kann man nun die vorher definierten Komponenten so platzieren, wie es das Projekt erfordert. Die Abmessungen des Boards werden nach Bedarf festgelegt und die Komponenten so angeordnet, dass bei den späteren Schritten des Lötens und der Verkabelung noch genug Arbeitsraum vorhanden ist. Das ist insbesondere bei Projekten wichtig, wo die Platine vielen Tests unterzogen wird und die Verbindungen oft gelöst und wieder befestigt werden müssen.

Im letzten Schritt wird das Layout und die Dicke der Leiterbahnen festgelegt. Da die Platine mithilfe eines Fräsers erstellt wird, ist es wichtig in der Darstellung des Boards nur einen Layer (Ebene) auszuwählen. Bei einfachen Projekten, bei denen man die Funktion „Autorouting“ verwenden will, also die automatische Verbindung der Komponenten, ist das umso wichtiger. Wird das nicht korrekt definiert, kann es vorkommen, dass die Leiterbahnen sich überscheiden, da das Programm von mehreren Layern ausgeht. Sind alle Einstellungen korrekt vorgenommen, inklusive der Dicke der Leiterbahnen, kann die Autoroutingfunktion verwendet werden und das für den Fräser benötigte monochrome Bild exportiert werden.

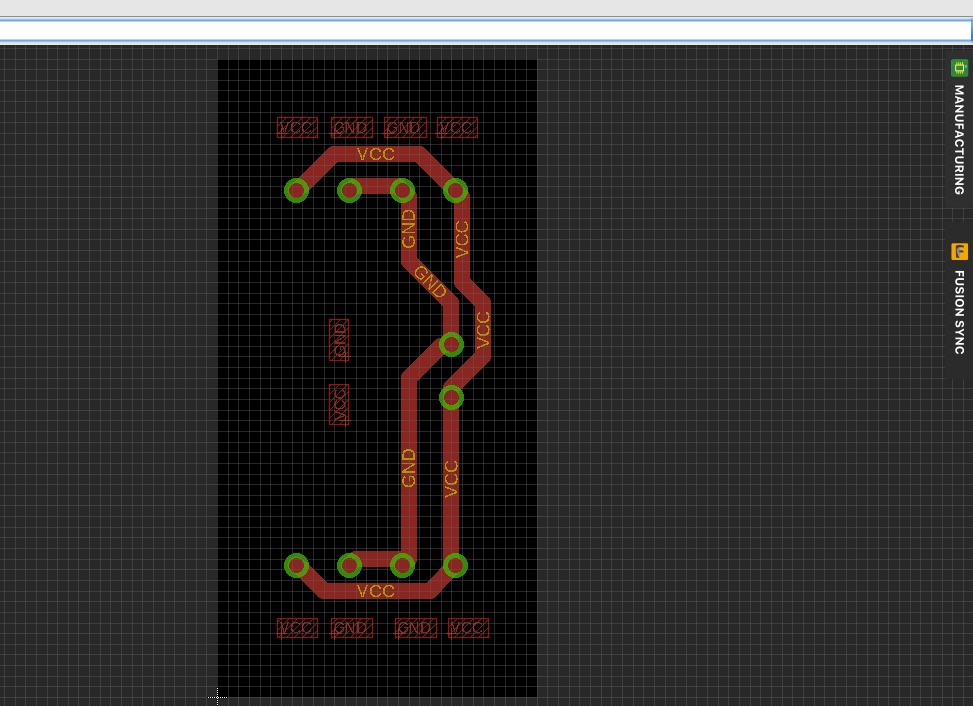


Abbildung 22: PCB-Board

Leider konnte das PDB der Last nicht standhalten und brannte nach wenigen Tests durch, weshalb Ersatz gekauft werden musste.

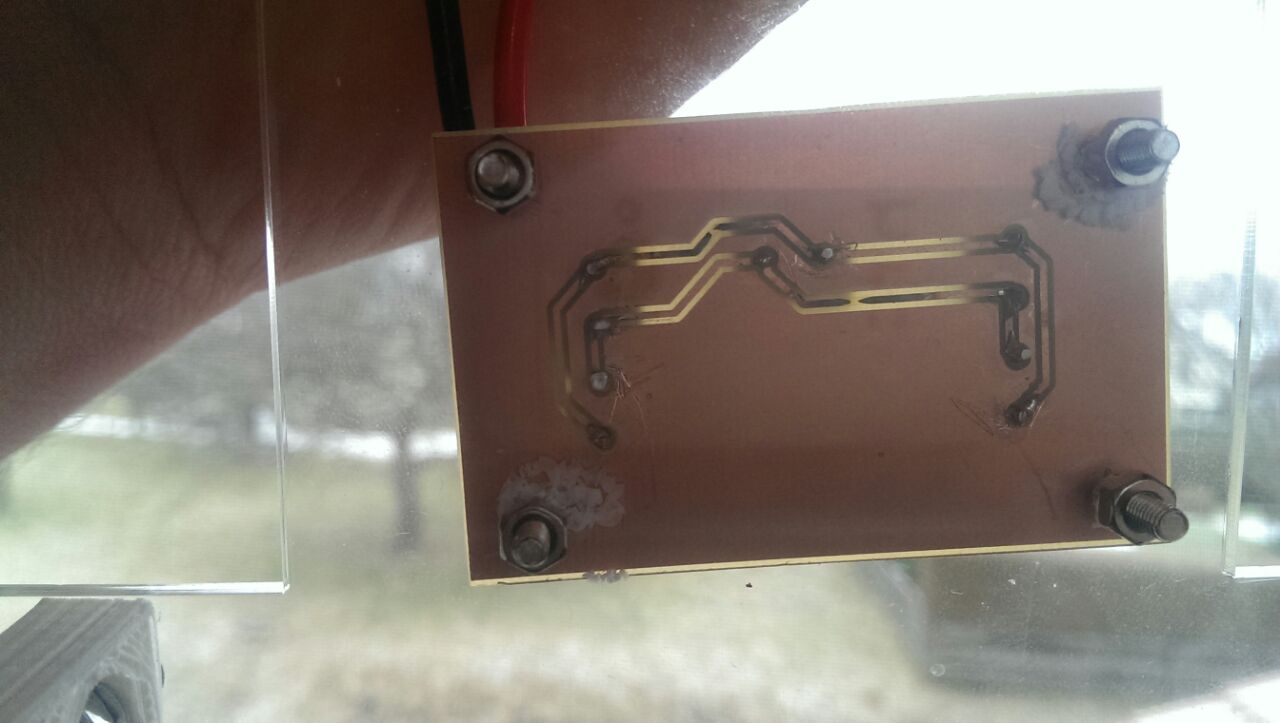


Abbildung 23: Durchgebranntes PCB

# 

# Lessons Learned

* Be realistic! Only do projects where scope, resources and time fits together.
* So, don’t be emotional! Don’t try thinks which are utopic only because you like them.
* As PM give clearly defined and small tasks. So that a team member can work step by step and not got overwhelmed.
* Everyone who completes a task has to document his doings and achieves instantly.